

Kapitel 3

Verteilungen II

3.1 *Monte-Carlo-Methode

Die Monte-Carlo-Methode benützt Pseudo-Zufallszahlen, die mit dem Computer erzeugt werden. Diese Methode ist sehr nützlich um ein Experiment zu simulieren und Daten zu erzeugen, die irgend eine Verteilung entsprechen. Im MATLAB 6 stehen u.a. folgende Funktionen zur Verfügung:

```
rand      - Uniform random numbers.
randn     - Normal (Gaussian) random numbers.
binornd   - Binomial random numbers.
poissrnd  - Poisson random numbers.
exprnd    - Exponential random numbers.
chi2rnd   - Chi square random numbers.
```

Im Folgenden werden Methoden, die zur Erzeugung von Monte-Carlo-Verteilungen benützt werden, besprochen.

3.1.1 Gleichverteilung und Gauss-Verteilung

Es gibt viele Methoden um Zufallszahlen zu erzeugen (siehe z.B. [Bran 99] für eine Einführung und [Pres 92], [Vatt 95] für einen kritischer Vergleich verschiedener Generatoren). Ein verbreiteter Algorithmus, der auch die MATLAB-Funktion `rand()` benützt, ist

$$x_{i+1} = a \times x_i \bmod m, \quad a = 7^5, \quad m = 2^{31} - 1 \quad (3.1)$$

Ausgehend von einem Startwert x_0 (*seed*), wird damit eine Serie von Zufallszahlen erzeugt, die alle möglichen Werte zwischen 1 und $m = 2147483647$ in willkürlicher

Reihfolge annehmen. m ist die maximale Integerzahl in einem 32-Bit Computer (Zahlenbereich Dezimal $-2147483648 \dots -1, 0, 1 \dots 2147483647$, Hexadezimal $80000000 \dots \text{FFFFFFFF}$, $0, 1 \dots 7\text{FFFFFFFF}$). Werden die mit dem Algorithmus 3.1 erzeugten x -Werte durch m dividiert, erhalten wir reelle Zufallszahlen, die zwischen 0 und 1 gleichverteilt sind.

MATLAB arbeitet mit “double precision” reellen Zahlen (64 Bits: Vorzeichen 1 Bit, Exponent 11 Bits und Mantisse 52 Bits; 52 Bits entsprechen einer Genauigkeit von mehr als 15 Dezimalziffern) und damit können wir den Algorithmus direkt programmieren und vergleichen mit der optimierten MATLAB-Funktion `rand()` (siehe Programm 3.1). Der Startwert $x_0 = 0$ ist nicht erlaubt. Wird die MATLAB-Funktion mit dem Startwert 0 aufgerufen (`rand('seed',0)`), so wird intern der Startwert auf 1144108930 gesetzt (dies ist auch der Default-Startwert beim Start von MATLAB). Damit sind die Zufallszahlen beim nächsten Programmlauf wieder gleich!

```
%-----
%   MATLAB program 3.1: random number generator
%   -----
rand('seed',0)
y = rand(10,1);           x = 1144108930;
a = 7^5;                  m = 2^31 -1;
fprintf('\n  i      x(i)      y(i)')
for i=1:10
    x = rem(a*x,m);        % x = a x mod m
    fprintf('\n%4.0f  %10.8f  %10.8f',i,x/m,y(i))
end
%-----
>> program31
      i      x(i)      y(i)
      1  0.21895919  0.21895919
      2  0.04704462  0.04704462
      3  0.67886472  0.67886472
      4  0.67929641  0.67929641
      5  0.93469290  0.93469290
      6  0.38350208  0.38350208
      7  0.51941637  0.51941637
      8  0.83096535  0.83096535
      9  0.03457211  0.03457211
     10  0.05346164  0.05346164
%-----
```

Daneben bietet MATLAB auch die Möglichkeit Zufallszahlen zu generieren, die der standardisierten Gauss-Verteilung $f(x;0,1)$ entsprechen (Algorithmen zur erzeugung normalverteilter Zufallszahlen werden in [Bran 99] beschrieben). Um Zufallszahlen zu erhalten, die einer beliebigen Gauss-Verteilung $f(x;\mu,\sigma)$ entsprechen, multipliziert man die erzeugten Zufallszahlen mit σ und addiert μ . Ein Beispiel ist in Programm 3.2 und Abb. 3.1 gegeben. Die Werte für Mittelwert, Varianz und Fehler des Mittelwerts entsprechen, sowohl für die Gleichverteilung als auch für die Gauss-Verteilung, den erwarteten Werten $(0.5, \sqrt{1/12}, \sqrt{1/12000}$; bzw. $0.5, 0.15, 0.15/\sqrt{1000}$).

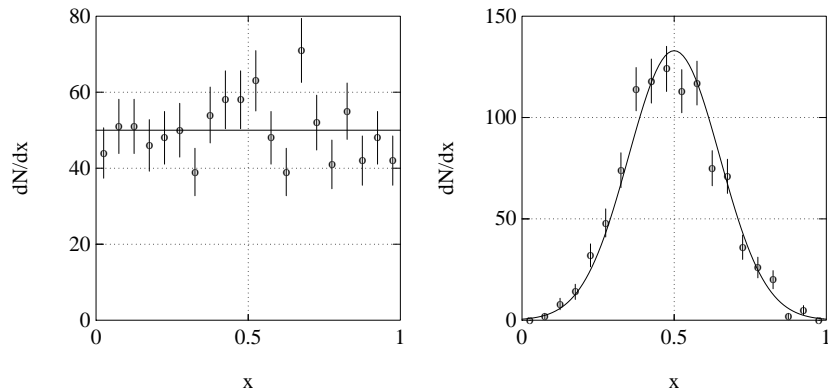


Abbildung 3.1: Monte-Carlo-Daten, die einer Gleichverteilung (links), bzw. einer Gauss-Verteilung (rechts) entsprechen.

```

%-----
%   MATLAB program 3.2: Uniform and Gaussian Monte Carlo Distributions
%   -----
rand('seed',0),                n = 1000;
data = rand(n,1);
bin = 0.05;                    x = bin/2:bin:1-bin/2;
subplot(121)
y = hist(data,x);              plot_data([x' y'])
xlabel('x'),                   ylabel('dN/dx')
axis([0 1 0 80]),             out = centroid(x,y)
plot([0 1],[50 50],'g-'),
randn('seed',0);

```

```

data = 0.15*randn(n,1) + 0.5;
subplot(122)
y = hist(data,x);              plot_data([x' y'])
axis([0 1 0 150]),           out = centroid(x,y)
x = 0:0.01:1;                 xlabel('x')
plot(x,n*bin*gauss(x,0.5,0.15),'g-'), ylabel('dN/dx')

```

```

%-----
% out =    0.4989    0.2800    0.0089
% out =    0.4975    0.1550    0.0049
%-----

```

3.1.2 Exponential- und Lorentz-Verteilung

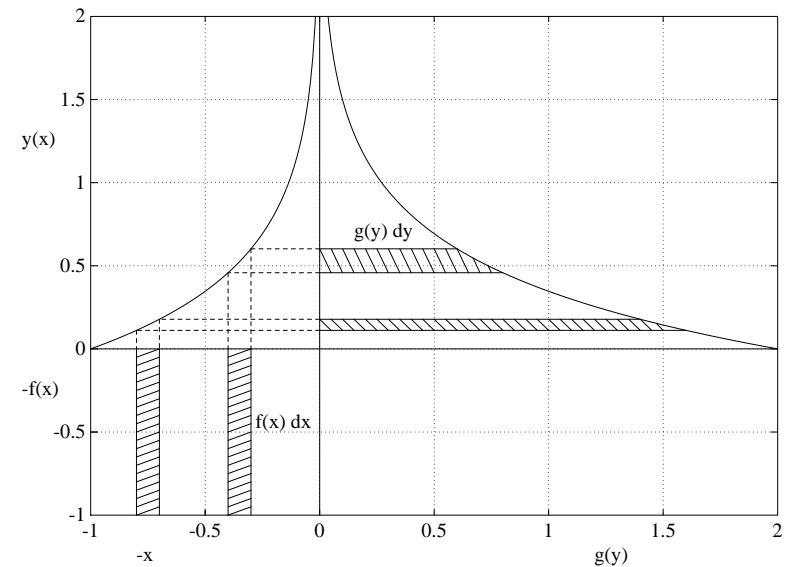


Abbildung 3.2: Variablentransformation $x \rightarrow y(x) = -0.5 \ln x$ und Transformation der Wahrscheinlichkeitsdichte $f(x) = 1 \rightarrow g(y) = 2 \exp(-2y)$.

Um beliebig verteilte Zufallszahlen zu erzeugen, brauchen wir eine Transformation der Gleichverteilung. Mit einer Variablentransformation $x \rightarrow y(x)$ ist auch eine

Transformation der Wahrscheinlichkeitsdichte $f(x) \rightarrow g(y)$ verbunden. Da die Wahrscheinlichkeit erhalten ist, gilt (Abb. 3.2)

$$f(x)dx = g(y)dy \quad (3.2)$$

Ist x gleichverteilt zwischen 0 und 1 ($f(x) = 1$), so gilt für die Wahrscheinlichkeitsfunktion $G(y)$

$$G(y) = \int_{y(0)}^{y(x)} g(y')dy' = \int_0^x dx' = x \quad (3.3)$$

$G(y)$ ist also gleichverteilt zwischen 0 und 1. Mit der Inversfunktion G^{-1} erhalten wir aus gleichverteilten Zufallszahlen x_i Zufallszahlen $y_i = G^{-1}(x_i)$, die der Verteilung $g(y)$ entsprechen. Dieses Verfahren ist nur anwendbar, wenn die Inversfunktion existiert. Für die Exponentialverteilung (Kap. 2.3)

$$g(y) = \lambda e^{-\lambda y} \quad (3.4)$$

ist dies der Fall (siehe Abb. 3.2). Aus

$$G(y) = \int_0^y \lambda e^{-\lambda y'} dy' = 1 - e^{-\lambda y} = x \quad (3.5)$$

folgt

$$\ln(1 - x) = -\lambda y$$

Auch $1 - x$ ist gleichverteilt zwischen 0 und 1. Der Algorithmus zur Erzeugung exponentialverteilter Zufallszahlen y_i ist also

$$y_i = \frac{-\ln x_i}{\lambda} \quad (3.6)$$

Ein Beispiel ist in Abb. 3.3 (Programm 3.3) gegeben. Die Verteilung ist abgeschnitten: oberhalb von $t = 6$ s liegen weitere 34 Ereignisse.

```
%-----
%   MATLAB program 3.3: Monte Carlo distributions
%   -----
%   ----- exponential distribution -----
lambda = 0.56;                n = 1000;
x = 0.1:0.2:6.1;             data = -log(rand(1,n))/lambda;
y = hist(data,x);            subplot(121)
plot_data([x' y']),          axis([0 6 0 120])
```

```
x = 0:0.1:8;
plot(x,200*lambda*exp(-lambda*x),'g-'),
xlabel('t [s]'),                ylabel('dN/dt [1/s]')

% ----- Lorentz distribution -----
subplot(122)
FWHM = 2;                        xmin = -10;
xmax = -xmin;                    bin = 1;
xb = xmin-bin/2:bin:xmax+bin/2;  z = tan(pi*(rand(n,1)-0.5));
y = hist(FWHM*z/2,xb);          logplot_data([xb' y'])
x = xmin:bin/5:xmax;           f = (FWHM/pi/2)./(x.^2 + FWHM^2/4);
semilogy(x,n*bin*f,'g-'),      axis([xmin xmax 1 1000]);
xlabel('E - E0 [MeV]'),        ylabel('dN/dE [1/MeV]')

%-----
% sum(f)/5 = 0.937,    sum(y(2:21)) = 929,    y(1) = 31,    y(22) = 40
%                    min(z) = -8129.6,    max(z) = 5.8928e+06
%-----
```

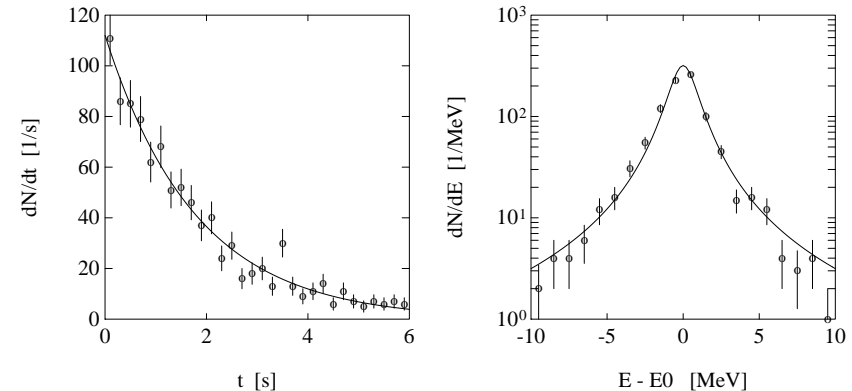


Abbildung 3.3: Monte-Carlo-Daten. Links: eine Exponentialverteilung; Rechts: eine Lorentz-Verteilung.

In der Mathematik ist die Lorentz-Verteilung als **Cauchy-Verteilung**

$$f(x) = \frac{1}{\pi(1+x^2)} \quad (3.7)$$

bekannt. Sie ist die Transformation $x = \tan \theta$ einer Gleichverteilung

$$g(\theta) = \frac{1}{\pi} \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$$

Mit $g(\theta) d\theta = f(x) dx$ (Abb. 3.2) erhält man

$$f(x) = \frac{1}{\pi} \frac{d\theta}{dx} = \frac{1}{\pi} \frac{\text{darctan} x}{dx} = \frac{1}{\pi} \frac{1}{1+x^2}$$

Mit $x = 2(E - E_0)/\Gamma$ und $f(x) dx = f(E) dE$ erhält man die Lorentz-Verteilung. Diese Transformationen werden benützt um Monte-Carlo-Daten, die einer Lorentz-Verteilung entsprechen, zu erzeugen. Ein Beispiel ist in Programm 3.3 (Abb. 3.3) gegeben. Im Intervall $(-10, 10) = (E_0 - 5\Gamma, E_0 + 5\Gamma)$ liegen nur 929 der 1000 erzeugten Ereignisse. Man beachte, dass alle Ereignisse mit $E < -10$ dem ersten Bin $y(1)$ und alle Ereignisse mit $E > 10$ dem letzten Bin $y(22)$ zugeordnet werden; deshalb wird das Histogramm vom Bin 2 bis Bin 21 gezeichnet.

3.1.3 Histogramm

Oft ist man nicht an einzelnen Ereignissen (Zufallszahlen, z.B. Zerfallszeiten) interessiert, sondern nur an histogrammierten Ereignissen (z.B. Zeitspektrum). In diesem Fall kann man mit Hilfe normalverteilter Zufallszahlen x_k (standardisierter Gauss-Verteilung) ein Histogramm mit statistischen Schwankungen erzeugt werden, durch die Zahl der zu erwartende Ereignisse N_k in Bin k zu "verschmieren"

$$N_k \rightarrow N_k + x_k \sqrt{N_k}.$$

Dies ist nur erlaubt, wenn jedes Bin eine genügende Anzahl Einträge enthält und damit die Poisson-Verteilung durch die Gauss-Verteilung ersetzt werden kann (siehe Kap. 2.2). Ein Beispiel ist in Abb. 3.4 (Programm 3.4) gegeben.

```

%-----
%   MATLAB program 3.4: Monte Carlo histogram
%   -----
%   gaussian peak with quadratic background
a = [220 -7.4 0.063 573 18.3 1.8];
k = 0:0.2:40; % plot theory
y = a(1) + a(2)*k + a(3)*k.^2 + a(4)*gauss(k,a(5),a(6));
plot(k,y,'b-')

```

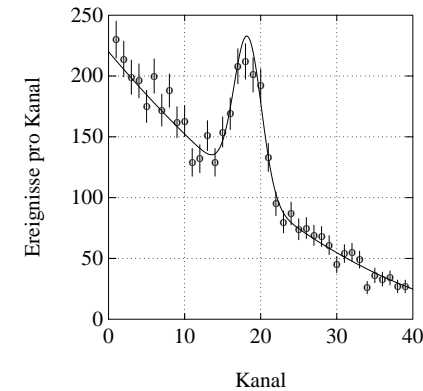


Abbildung 3.4: Monte-Carlo-Histogramm: Gauss-Peak mit quadratischem Untergrund.

```

k = 1:39; % plot Monte Carlo data
y = a(1) + a(2)*k + a(3)*k.^2 + a(4)*gauss(k,a(5),a(6));
randn('seed',0)
y = y + sqrt(y).*randn(1,length(k)); y = round(y);
hold on, plot_data([k' y'])
grid
xlabel('Kanal'), ylabel('Ereignisse pro Kanal')
%-----

```

3.1.4 Integration; Phasenraumverteilungen

Die Monte-Carlo-Methode wird auch benützt um mehrdimensionale Integrale auszurechnen. Das Prinzip wollen wir an einem einfachen Beispiel, der Quadratur des Kreises, demonstrieren. Zur Berechnung der Kreisfläche werden N gleichverteilte Punkte (x_i, y_i) (x_i und y_i sind unkorrelierte gleichverteilte Zufallszahlen) im kreiseinschliessenden Viereck erzeugt (Abb. 3.5, Programm 3.5). Das Verhältnis Kreisfläche / Vierecksfläche = Nk/N , wobei Nk die Zahl der Punkte in der Kreisfläche ist. Damit lässt sich π berechnen. Die Genauigkeit hängt von der Zahl der generierte Punkten ab (Binomialverteilung, Kap. 2.1). Für $N = 1000$ ist der Fehler 0.05 (Output von Programm 3.5).

Mit dieser Methode, die einen Teil der generierten Punkte selektiert (Von Neumanns "accept-reject"-Methode), kann man auch Zufallszahlen, die einer beliebigen Verteilung entsprechen, erzeugen. Im Beispiel (Abb. 3.5, Programm 3.5) entspricht die x -Koordinate der Punkte in der Kreisfläche, einer Verteilung

$$f(x) = \frac{4}{\pi} \sqrt{1-x^2}, \quad 0 \leq x \leq 1$$

Man sehe sich im Programm 3.5 genau an, wie im MATLAB eine Selektion gemacht wird!

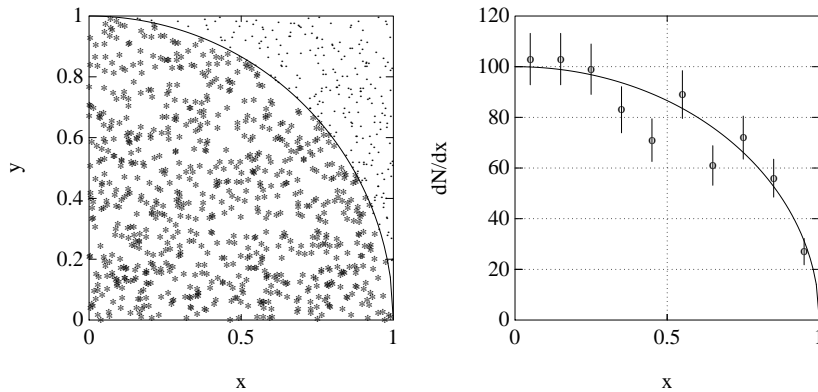


Abbildung 3.5: Monte-Carlo-Integration. Links: Fläche des Kreissegments; Rechts: die dazugehörige x -Verteilung.

```

%-----
%   MATLAB program 3.5: Monte Carlo integration: calculation of pi
%   -----
n = 1000;
x = rand(1,n);
r = sqrt(x.^2 + y.^2);
MCpi = 4*sum(accept)/n,
subplot(121),
plot(x(~accept),y(~accept),'b.'),
plot(x(accept),y(accept),'r*'),
plot(z,sqrt(1-z.^2),'k-'),
axis square
xlabel('x'),
y = rand(1,n);
accept = r <= 1;
s = sqrt(MCpi*(4-MCpi)/n)
hold on
z = 0:0.01:1;
axis([0 1 0 1])
ylabel('y')

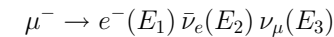
```

```

subplot(122)
z = 0.05:0.1:0.95;
subplot(122),
plot_data([z' Nz']),
z = 0:0.01:1;
axis square
xlabel('x'),
Nz = hist(x(accept),z);
axis([0 1 0 120])
plot(z,n*0.1*sqrt(1-z.^2),'k-')
ylabel('dN/dx')
%-----
%   MCpi = 3.20, 3.15, 3.03, 3.13, 3.19; sigma = 0.05
%-----

```

Mehrdimensionale Integrale kommen in der Teilchenphysik oft vor. In einer Reaktion mit vielen Teilchen im Endzustand können oft nur ein Teil der physikalisch relevanten Observablen gemessen werden. Zum Vergleich mit der theoretische Vorhersage wird über die nicht beobachteten Variablen integriert. Ein einfaches Beispiel ist der Myonzerfall



Der Impulsphasenraum eines Dreikörperzerfalls hat nur zwei physikalisch relevante Observablen, z.B. die Energie des Elektrons E_1 und die Energie des Antineutrinos E_2 . Die Phasenraumdichte im Dalitzplot ist konstant (siehe Vorlesung Kern- und Teilchenphysik I)

$$\frac{d^2 R_3}{dE_1 dE_2} = \frac{1}{32\pi^3}$$

Für die drei masselosen Teilchen im Myonzerfall ($m_\mu \gg m_e \simeq 0$) ist die minimale Energie $E_{min} = 0$ und die maximale Energie $E_{max} = m_\mu c^2/2$. Der Dalitzplot (2-dimensionale Phasenraumverteilung) ist mit dem Monte-Carlo-Methode berechnet worden (Programm 3.6) und in Abb. 3.6 dargestellt, wobei die relativen Energien $x = 2E_1/E_{max}$ und $y = 2E_2/E_{max}$ benützt wurden. Aus der Kinematik folgt, dass $x + y \geq 1$ ist. Durch Integration über y

$$\frac{dR_3}{dx} \propto \int_{1-x}^1 dy = x$$

erhält man die 1-dim. Phasenraumverteilung des Elektrons. In Abb. 3.6 ist diese Funktion mit den histogrammierten Monte-Carlo-Daten verglichen.

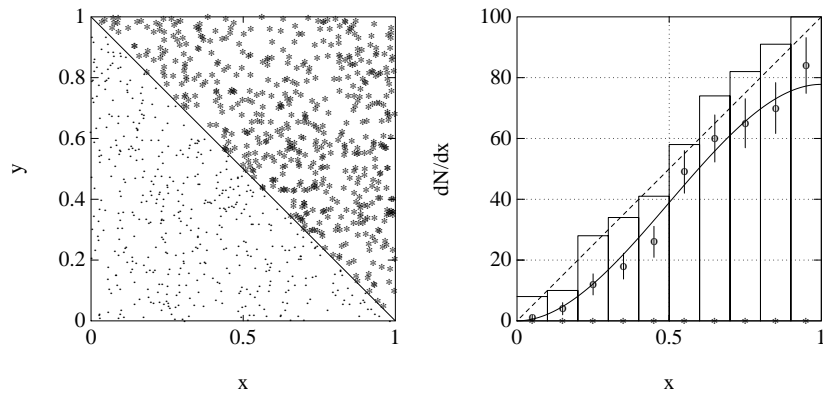


Abbildung 3.6: Monte-Carlo-Daten des Myonzerfalls. Links: 2-dimensionale Phasenraumverteilung; Rechts: die dazugehörige x -Verteilung mit (—, Punkthistogramm) und ohne Matricelement (- -, Balkenhistogramm).

Die Phasenraumdichte multipliziert mit dem Matricelement $M \propto y(1-y)$ (Fermis Goldene Regel) und integriert über y ergibt das Energiespektrum des Elektrons

$$\frac{d\Gamma_{\mu}}{dx} \propto \int_{1-x}^1 y(1-y)dy = x^2/2 - x^3/3$$

in Übereinstimmung mit Gl. 1.6 für $\rho = 3/4$.

In der Monte-Carlo-Simulation wird das Matricelement mit dem “accept-reject”-Methode berücksichtigt. Die Phasenraumereignisse werden mit einer Wahrscheinlichkeit, die dem Matricelement proportional ist, akzeptiert (siehe Programm 3.6). Das Resultat ist in Abb. 3.6 mit der analytischen Funktion verglichen.

```

%-----
%   MATLAB program 3.6: Monte Carlo phase space: muon decay
%   -----
n = 1000;
x = rand(1,n);
accept = x+y >= 1;
xa = x(accept);
subplot(121)
xlabel('x'),
plot(x(~accept),y(~accept),'b.'),
plot(xa,ya,'r*'),
axis square
plot([0 1],[1 0],'k-')
z = 0.05:0.1:0.95;
subplot(122)
b = bar(z, Nz,1);
hold on
plot([0 1],[0 0.1*n],'b--'),
xlabel('x'),
me = ya.*(1-ya);
accept = me >= rand(1,length(ya));
Nz = hist(xa(accept),z);
z = 0:0.01:1;
axis square
%-----
y = rand(1,n);
ya = y(accept);
ylabel('y')
hold on
axis([0 1 0 1])
Nz = hist(xa,z);
set(b,'FaceColor','y')
axis([0 1 0 100])
ylabel('dN/dx')
me = me/max(me);
N = sum(accept)
plot_data([z' Nz'])
plot(z,N*0.1*(6-4*z).*z.^2,'b-')

```

3.2 Zweidimensionale Gauss-Verteilung

3.2.1 Theorie und *Monte-Carlo-Beispiel

Die eindimensionale Gauss-Verteilung ist (Standardform mit $\mu = 0$ und $\sigma = 1$, siehe Kap. 1.4)

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Die Gleichung für eine konstante Wahrscheinlichkeitsdichte ist damit $x^2 = k^2$ (zwei Punkte). Die Wahrscheinlichkeit einen x -Wert mit $x^2 \leq k^2$ zu beobachten ist

$$\int_{-k}^k f(x) dx = 68.3\% \text{ (95.4\%, 99.7\%)} \quad \text{für } k = 1 \text{ (2, 3)}$$

Sind x und y zwei *unkorrelierte* normalverteilte Zufallsvariable, dann ist die gemeinsame Wahrscheinlichkeitsdichte eine zweidimensionale Gauss-Verteilung

$$f(x, y) = f(x)f(y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2} \quad (3.8)$$

Nur für unkorrelierte Variablen ist die Wahrscheinlichkeit $f(x, y) dx dy$, gleichzeitig ein x -Wert im Intervall $(x, x + dx)$ und ein y -Wert im Intervall $(y, y + dy)$ zu beobachten, gleich dem Produkt der Einzelwahrscheinlichkeiten $f(x) dx f(y) dy$. Die Gleichung einer konstanten Wahrscheinlichkeitsdichte ist $x^2 + y^2 = k^2$ (Kreis). Die Wahrscheinlichkeit ein x, y -Paar mit $x^2 + y^2 \leq k^2$ zu beobachten ist (Polarkoordinaten: $x = r \cos \phi$, $y = r \sin \phi$)

$$\begin{aligned} \int_0^{2\pi} \int_0^k \frac{1}{2\pi} e^{-r^2/2} d\phi r dr &= \int_0^k e^{-r^2/2} d(r^2/2) = 1 - e^{-k^2/2} \\ &= 39.4\% \text{ (86.5\%, 98.9\%)} \quad \text{für } k = 1 \text{ (2, 3)} \end{aligned} \quad (3.9)$$

Die allgemeine Form der zweidimensionalen Gauss-Verteilung ist

$$\boxed{f(x, y) = \frac{1}{2\pi \sigma_x \sigma_y \sqrt{1 - \rho^2}} \times \exp\left(\frac{-1}{2(1 - \rho^2)} \left[\frac{(x - \mu_x)^2}{\sigma_x^2} - 2\rho \frac{(x - \mu_x)(y - \mu_y)}{\sigma_x \sigma_y} + \frac{(y - \mu_y)^2}{\sigma_y^2} \right]\right)} \quad (3.10)$$

wobei ρ der Korrelationskoeffizient ist ($\rho = 0$ für unkorrelierte Variablen). Die Gleichung einer konstanten Wahrscheinlichkeitsdichte ist

$$\frac{(x - \mu_x)^2}{\sigma_x^2} - 2\rho \frac{(x - \mu_x)(y - \mu_y)}{\sigma_x \sigma_y} + \frac{(y - \mu_y)^2}{\sigma_y^2} = k^2(1 - \rho^2) \quad (3.11)$$

Dies ist die Gleichung einer Ellipse, die sogenannte Kovarianzellipse. Die **Kovarianz** zweier Variablen ist

$$\boxed{\text{cov}(x, y) \equiv \langle (x - \mu_x)(y - \mu_y) \rangle = \langle xy \rangle - \mu_x \mu_y} \quad (3.12)$$

Zur Berechnung der Kovarianz transformieren wir die Variablen x und y auf die Standardform

$$u = \frac{x - \mu_x}{\sigma_x} \quad v = \frac{y - \mu_y}{\sigma_y} \quad (3.13)$$

und mit $f(u, v) dudv = f(x, y) dx dy$ erhält man

$$f(u, v) = \frac{1}{2\pi \sqrt{1 - \rho^2}} \exp\left(\frac{-1}{2(1 - \rho^2)} [u^2 - 2\rho uv + v^2]\right) \quad (3.14)$$

Mit

$$u^2 - 2\rho uv + v^2 = (u - \rho v)^2 + (1 - \rho^2)v^2$$

erhält man für die Kovarianz der Variablen u und v

$$\begin{aligned} \langle uv \rangle &= \int \int uv f(u, v) dudv \\ &= \frac{1}{2\pi \sqrt{1 - \rho^2}} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \exp\left(\frac{-(u - \rho v)^2}{2(1 - \rho^2)}\right) u du \right] \exp\left(\frac{-v^2}{2}\right) v dv \\ &\quad \text{(Gauss-Verteilung mit } \langle u \rangle = \rho v \text{ und } \sigma^2 = (1 - \rho^2)) \\ &= \frac{\rho}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(\frac{-v^2}{2}\right) v^2 dv = \rho \quad (\langle v^2 \rangle = 1) \end{aligned}$$

Die **Korrelationskoeffizient** ist also

$$\boxed{\rho = \langle uv \rangle = \left\langle \frac{x - \mu_x}{\sigma_x} \frac{y - \mu_y}{\sigma_y} \right\rangle = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}} \quad (3.15)$$

Die Gleichung der **Kovarianzellipse** in der Standardform ist

$$\boxed{u^2 - 2\rho uv + v^2 = k^2(1 - \rho^2)} \quad (3.16)$$

Mit der Transformation (45°-Drehung)

$$s = (u + v)/\sqrt{2} \quad t = (v - u)/\sqrt{2}$$

erhält man

$$s^2(1 - \rho) + t^2(1 + \rho) = k^2(1 - \rho^2)$$

oder

$$\frac{s^2}{(1 + \rho)} + \frac{t^2}{(1 - \rho)} = k^2 \quad (3.17)$$

Dies ist die Gleichung einer Ellipse mit den Halbachsen

$$r_1 = \sqrt{(1 + \rho)} k \quad \text{und} \quad r_2 = \sqrt{(1 - \rho)} k \quad (3.18)$$

Die minimale und maximale Werte von ρ sind $\rho_{min} = -1$ und $\rho_{max} = +1$. Für diese Werte ist die Ellipse in eine Gerade entartet (maximale Korrelation $v = -u$ bzw. $v = u$). Für $\rho = 0$ ist die Ellipse ein Kreis und die Variablen u und v sind unkorreliert.

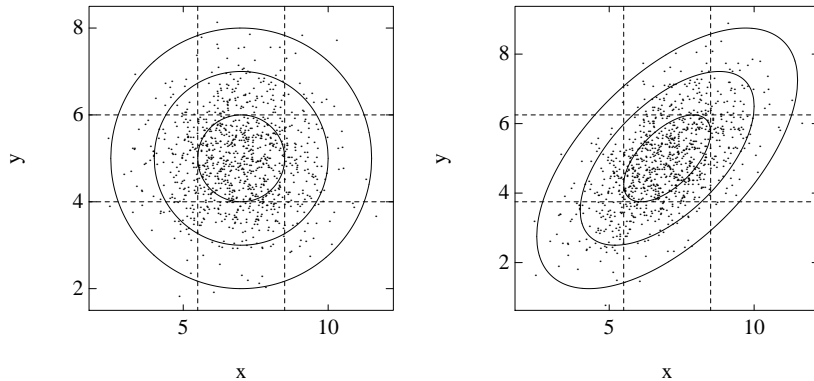


Abbildung 3.7: Zweidimensionale Gauss-Verteilungen: Monte-Carlo-Daten mit den 1σ -, 2σ -, und 3σ -Kovarianzellipsen. Links: $\rho = 0$, rechts: $\rho = 0.6$. Die y -Skala ist so gewählt, dass σ_x und σ_y die gleiche Länge in der Figur haben.

Ein ***Monte-Carlo-Beispiel** ist im Programm 3.7 und Abb. 3.7 gegeben. Der Algorithmus zu Erzeugung korrelierter (x_i, y_i) -Paare ist

$$x_i = \sigma_x z_{1i} + \mu_x \quad y_i = \sigma_2 z_{2i} + \mu_y + c(x_i - \mu_x)$$

wobei z_{1i} und z_{2i} zwei unkorrelierte normalverteilte ($\mu_z = 0$, $\sigma_z = 1$) Variable sind. Die Varianz von y ist (Gl. 1.31)

$$\sigma_y^2 \equiv \frac{1}{n} \sum_{i=1}^n (y_i - \mu_y)^2 = \frac{1}{n} \sum_{i=1}^n [\sigma_2 z_{2i} + c(x_i - \mu_x)]^2 = \sigma_2^2 + c^2 \sigma_x^2$$

und die Kovarianz

$$\begin{aligned} \text{cov}(x, y) &\equiv \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)[\sigma_2 z_{2i} + c(x_i - \mu_x)]^2 \\ &= \frac{c}{n} \sum_{i=1}^n (x_i - \mu_x)^2 = c \sigma_x^2 \end{aligned}$$

Das Beispiel wurde mit folgenden Parameter generiert

$$\begin{aligned} \mu_x = 7 \quad \sigma_x = 1.5 \quad \mu_y = 5 \quad \sigma_2 = 1 \quad \text{und} \quad \rho = 0 \quad (\sigma_y = 1) \\ \text{oder} \quad \rho = 0.6 \quad (\sigma_y = 1.25) \end{aligned}$$

Die “gemessenen” Werte sind damit in Übereinstimmung (siehe Programm 3.7). Für $n = 1000$ generierte (x_i, y_i) -Paare erwartet man $np \pm \sqrt{np(1-p)}$ (Binomialverteilung) innerhalb der $k\sigma$ -Kovarianzellipse (p ist in Gl. 3.9 gegeben; bei der Transformation von Kreis zu Ellipse bleibt die Wahrscheinlichkeit erhalten)

$$k = 1 : 394 \pm 15 \quad k = 2 : 865 \pm 11 \quad k = 3 : 989 \pm 3$$

Die “beobachten” Zahlen sind damit in Übereinstimmung.

```

%-----
%   MATLAB program 3.7: 2 dim. Gauss distribution
%   -----
n = 1000;
mx = 7;                               sx = 1.5;
my = 5;                               sy = 1.0;
x = sx*randn(n,1)+mx;                 y = sy*randn(n,1)+my;
c = 0.5;                               y = y + c*(x - mx);   % c = 0;
sy = sqrt(sy^2 + c^2*sx^2),           rho = c*sx/sy,
xmean = mean(x),                      sigmax = std(x)
ymean = mean(y),                      sigmay = std(y)
plot(x,y,'r*'),                       hold on
axis([mx-3.5*sx mx+3.5*sx my-3.5*sy my+3.5*sy])
u = (x - mx)/sx;                       v = (y - my)/sy;
r = sqrt((u.^2 - 2*rho*u.*v + v.^2)/(1-rho^2));

```



```

rhoxy = sum(u.*v)/n
for k = 1:3
    xk = -k:0.01:k;
    yk = sy*(sqrt((k^2 - xk.^2)*(1-rho^2)) + rho*xk) + my;
    plot(mx + sx*xk,yk,'b-'),      plot(mx - sx*xk,-yk+2*my,'b-')
    select = r <= k;                nk = sum(select)
end
plot([mx-sx mx+sx],[0 10],'b--'), plot([mx+sx mx+sx],[0 10],'b--')
plot([0 15],[my-sy my+sy],'b--'), plot([0 15],[my+sy my+sy],'b--')
xlabel('x'),                          ylabel('y')
axis square

%-----
% xmean = 7.01, sigmax = 1.55, ymean = 5.07, sigmay = 0.98, rho = -0.01
% n1 = 408,      n2 = 861,      n3 = 987
% xmean = 6.98, sigmax = 1.49, ymean = 4.93, sigmay = 1.29, rho = +0.61
% n1 = 379,      n2 = 857,      n3 = 986
%-----

```

3.2.2 Normalverteilte Fehler und χ^2

Ein typisches Beispiel von korrelierten Variablen ist die Anpassung einer linearen Funktion $f(x) = ax + b$ an gemessene Daten $(x_i, y_i \pm \sigma_i, i = 1 \dots n)$, wobei die Parameter (Variablen) a und b meistens stark korreliert sind. Ein Beispiel ist in Abb. 3.8 (Programm 3.8) gegeben. χ^2 ist eine Funktion von a und b (Gl.2.23)

$$\chi^2(a, b) = \sum_{i=1}^n \frac{[y_i - f(x_i; a, b)]^2}{\sigma_i^2} \quad (3.19)$$

Das Minimum $\chi^2(a_0, b_0)$ von $\chi^2(a, b)$ kann graphisch ermittelt werden (Abb. 3.8). In Analogie zur eindimensionalen χ^2 -Funktion (Kap. 1.3.5) werden die Fehler $k\sigma_a$ und $k\sigma_b$ aus den Tangenten an die $k\sigma$ -**Kovarianzellipse**

$$\chi^2(a, b) = \chi^2(a_0, b_0) + k^2 \quad (3.20)$$

abgelesen. Der Korrelationskoeffizient kann am genauesten aus der Länge der kurzen Halbachse (Gl. 3.18) abgelesen werden

$$\rho = r_1^2/k^2 - 1 \quad (\rho < 0) \quad \text{oder} \quad \rho = 1 - r_2^2/k^2 \quad (\rho > 0)$$

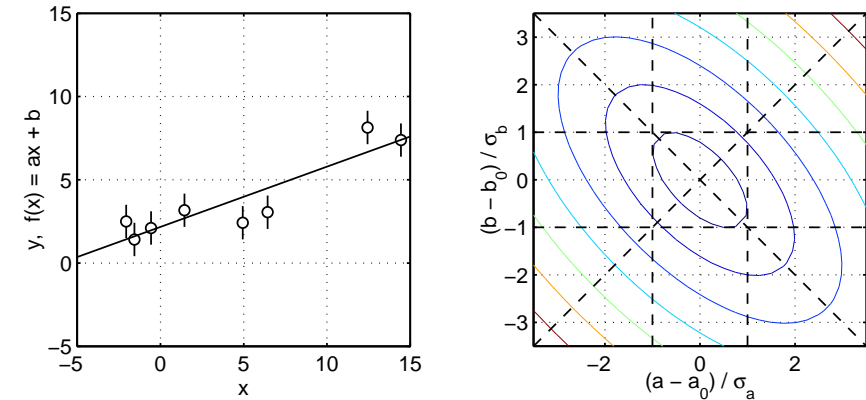


Abbildung 3.8: Anpassung einer linearen Funktion mit der Methode der kleinsten Quadrate. Links: Monte-Carlo-Daten $(x_i, y_i \pm \sigma_i)$ und Fitfunktion $f(x) = ax + b$. Rechts: Höhenliniendarstellung der $\chi^2(a, b)$ -Funktion

Der Vergleich der zweidimensionalen Gauss-Verteilung (Abb. 3.7) mit der $\chi^2(a, b)$ -Funktion (Abb. 3.8) zeigt eindrücklich, dass die Fehler (statistische Schwankungen) gaussisch verteilt sind. Damit können wir folgende Aussage der Statistik machen (Gl. 3.9):

Mit 39.4 % Wahrscheinlichkeit schliesst die 1σ -Kovarianzellipse den "wahren" Punkt $(\langle a \rangle, \langle b \rangle)$ ein.

Sind wir nur an einem der beiden Parameter interessiert, z.B. a , können wir die gleiche Aussage machen wie bei einem Parameter:

Mit 68.3 % Wahrscheinlichkeit schliesst das 1σ -Intervall $(a_0 - \sigma_a, a_0 + \sigma_a)$ den "wahren" Wert $\langle a \rangle$ ein.

In Programm 3.8 wird die zweidimensionale $\chi^2(a, b)$ -Funktion mit Höhenlinien dargestellt (Kontourplot). Die wichtigsten MATLAB-Befehle dazu sind `[a,b] = meshgrid(a,b)` und `contour((a-a0)/sa,(b-b0)/sb,chi2,v)`. Mit dem ersten Befehl wird ein Gitter von Stützpunkten bestimmt, wo die Funktion `chi2` berechnet wird. Mit dem zweiten Befehl wird der Kontourplot gemacht, wobei die Abstufung der Höhenlinien im Vektor `v = (1:10).^2 + min(min(chi2))` gegeben sind und die x - und y -Koordinaten in den Matrizen $(a-a_0)/s_a$ und $(b-b_0)/s_b$.

```

%-----
%  program 3.8: contour plot of chi^2; linear fit f = ax + b
%  -----
subplot(121)
load linearData;
x = linearData(:,1);
y = linearData(:,2);
plot_data(linearData),
grid,
xlabel('x'),
a0 = 0.362;
sa = 0.060;
plot([-5 15],a0*[-5 15]+b0,'g-'),
sigma = -3.5:0.25:3.5;
a = sa*sigma + a0;
[a,b] = meshgrid(a,b);
for i = 1:length(x)
    chi2 = chi2 + (y(i) - (a*x(i)+b)).^2/sig(i)^2;
end
v = (1:10).^2 + min(min(chi2));
contour((a-a0)/sa,(b-b0)/sb,chi2,v),
grid,
xlabel('(a - a_0) / \sigma_a'),
plot([-3.5 3.5],[3.5 -3.5],'k--'),
fprintf('Specify short half axis of 3 sigma ellips\n')
[x3, y3] = ginput(1),
rho = (x3^2 + y3^2)/9 - 1

%-----
%  rho = -0.59
%-----

```

3.2.3 Kovarianzmatrix und Fehlerfortpflanzungsgesetz

Die Erweiterung auf mehr als zwei Dimensionen ($n > 2$) ist einfach, wenn wir eine formale Notation benützen. Statt der zwei Variablen (x, y) mit den Erwartungswerten (μ_x, μ_y) haben wir die n Variablen $\vec{x}^T = (x_1, \dots, x_n)$ (\vec{x} ist ein Spaltenvektor und der Transponiertenvektor \vec{x}^T ein Zeilenvektor) mit den Erwartungswerten $\vec{\mu}^T = (\mu_1, \dots, \mu_n)$. Damit wird die n -dimensionale Gauss-Verteilung

$$f(\vec{x}) = \frac{1}{\sqrt{(2\pi)^n |C|}} \exp \left[-\frac{1}{2} (\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right] \quad (3.21)$$

wobei C die $n \times n$ **Kovarianzmatrix** ist

$$C_{ii} = \sigma_i^2 \quad C_{ij} = C_{ji} = \text{cov}(x_i, x_j) = \rho_{ij} \sigma_i \sigma_j \quad (3.22)$$

und $|C|$ die Determinante von C . Explizit ausgeschrieben für $n = 2$ Dimensionen

$$C = \begin{pmatrix} \sigma_1^2 & \rho \sigma_1 \sigma_2 \\ \rho \sigma_1 \sigma_2 & \sigma_2^2 \end{pmatrix} \quad C^{-1} = \frac{1}{|C|} \begin{pmatrix} \sigma_2^2 & -\rho \sigma_1 \sigma_2 \\ -\rho \sigma_1 \sigma_2 & \sigma_1^2 \end{pmatrix}$$

und $|C| = \sigma_1^2 \sigma_2^2 (1 - \rho^2)$. Dies in Gl. 3.21 eingesetzt ergibt Gl. 3.10.

Auch das Fehlerfortpflanzungsgesetz ist komplizierter für korrelierte Variablen. Zuerst wollen wir ein einfaches Beispiel, die Anpassung einer linearen Funktion $f(x; a, b) = ax + b$ an Messdaten (Abb. 3.8), betrachten. Die optimale Lösung ist (χ^2 minimal) $f_0(x) \equiv f(x; a_0, b_0)$. Damit können wir für jeden x -Wert den optimalen Schätzwert $f_0(x)$ berechnen (lineare Interpolation oder Extrapolation). Der Fehler dieses Schätzwertes ist

$$\begin{aligned} \sigma_f^2 &= \langle (f - f_0)^2 \rangle = \langle (ax + b - a_0x - b_0)^2 \rangle \\ &= \langle x^2(a - a_0)^2 + (b - b_0)^2 + 2x(a - a_0)(b - b_0) \rangle \\ &= x^2 \sigma_a^2 + \sigma_b^2 + 2x \text{cov}(a, b) \end{aligned}$$

Die allgemeine Form des **Fehlerfortpflanzungsgesetz** für eine Funktion von n korrelierten Variablen $f(a_1, \dots, a_n)$ ist

$$\sigma_f^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial a_i} \right)^2 \sigma_{a_i}^2 + \sum_{i=1}^n \sum_{j \neq i}^n \left(\frac{\partial f}{\partial a_i} \frac{\partial f}{\partial a_j} \right) \text{cov}(a_i, a_j) \quad (3.23)$$

Das naive Fehlerfortpflanzungsgesetz (ohne die Kovarianzen) gilt also nur für unabhängige Variablen! Man beachte, dass in der Doppelsumme jedes Paar zweimal vorkommt, (a_i, a_j) und (a_j, a_i) . Das Fehlerfortpflanzungsgesetz gilt allgemein für Funktionen, die linear von den Variablen abhängen, da in diesem Fall die Ableitungen konstant sind. Für nicht lineare Funktionen gilt es nur für kleine Fehler σ_a , da man dann die Taylor-Entwicklung

$$f(a) = f(a_0) + (a - a_0) \left(\frac{\partial f}{\partial a} \right)_{a=a_0} + \dots \quad (3.24)$$

bei der ersten Ordnung abbrechen kann. Formal kann man das Fehlerfortpflanzungsgesetz auch mit der Kovarianzmatrix schreiben

$$\sigma_f^2 = \vec{d}^T C \vec{d} \quad \vec{d} = \left(\frac{\partial f}{\partial a_1}, \dots, \frac{\partial f}{\partial a_n} \right) \quad (3.25)$$

Für unser Beispiel ist das $\text{cov}(a, b) = \rho \sigma_a \sigma_b$

$$\sigma_f^2 = (x, 1) \begin{pmatrix} \sigma_a^2 & \rho \sigma_a \sigma_b \\ \rho \sigma_a \sigma_b & \sigma_b^2 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = x^2 \sigma_a^2 + \sigma_b^2 + 2x \rho \sigma_a \sigma_b$$

Bildet man nicht nur eine, sondern m Funktionen f_j der n Variablen (a_1, \dots, a_n) , so ist man neben den Fehlern $\sigma_{f_j}^2 = C_{jj}^f$ auch an die Kovarianzen $\text{cov}(f_j, f_k) = C_{jk}^f$ interessiert. Das **Fehlerfortpflanzungsgesetz** ist in diesem allgemeinen Fall

$$C^f = D^T C D \quad D_{ij} = \frac{\partial f_j}{\partial a_i} \quad (3.26)$$

wobei C die $n \times n$ -Kovarianzmatrix der Variablen ist, D eine $n \times m$ -Matrix der partielle Ableitungen (D^T ist eine $m \times n$ -Matrix) und C^f die $m \times m$ -Kovarianzmatrix der Funktionen.

3.3 Faltung von Verteilungen

Eine Messgrösse z (z.B. eine Zeit $z = t$) ist die Summe der physikalisch relevanten Grössen x (z.B. die Zerfallszeit $x = t_1$) und eines Messfehlers y ($y = t_2$). Die Verteilungen der physikalischen Grössen $g(x)$ und des Messfehlers $h(y)$ sind bekannt (z.B. Exponential- und Gauss-Verteilung). Gefragt ist die Verteilung der Messgrösse $f(z)$. Dies ist die Faltung (“convolution”) zweier Verteilungen

$$\boxed{z = x + y} \quad \boxed{f(z) = g(x) \otimes h(y)} \quad (3.27)$$

Die Faltung zweier diskreter Verteilungen ist in der numerischen Datenanalyse sehr wichtig, da auch kontinuierliche Verteilungen nur an einer endlichen Zahl von Stützpunkten berechnet werden. Von theoretischem Interesse ist, neben der Faltung zweier Poisson-Verteilungen, vor allem die Faltung zweier kontinuierlicher Verteilungen, wie z.B. die Faltung einer Exponentialverteilung mit einer Gauss-Verteilung.

3.3.1 Diskrete Verteilungen

Die beiden primären Grössen sind

$$\begin{aligned} x_i & \quad i = 1 \dots m & \Delta x = x_i - x_{i-1} = \Delta \\ y_j & \quad j = 1 \dots n & \Delta y = y_j - y_{j-1} = \Delta \end{aligned} \quad (3.28)$$

mit den Verteilungen

$$g(x_i) \equiv g_i \quad \text{und} \quad h(y_j) \equiv h_j \quad (3.29)$$

Man beachte, dass die Binbreite Δ für beide Verteilungen gleich und konstant ist. Die möglichen z -Werte sind

$$z_k = x_i + y_j \quad \Rightarrow \quad z_1 = x_1 + y_1 \dots z_{m+n-1} = x_m + y_n \quad (3.30)$$

Die Wahrscheinlichkeit $f(z_k) \equiv f_k$ einen Wert $z_k = x_i + y_j$ zu beobachten hängt ab von der Anzahl der Kombinationen i, j mit dem gleichen Wert $k = i + j - 1$ und von den einzelnen Wahrscheinlichkeiten g_i und h_j

$$\boxed{f_k = \sum_{i=\max(1, k-n+1)}^{\min(k, m)} g_i h_{k+1-i} = \sum_{j=\max(1, k-m+1)}^{\min(k, n)} g_{k+1-j} h_j} \quad (3.31)$$

Zur Verdeutlichung seien einige Werte explizit ausgeschrieben

$$\begin{aligned} k = 1, & \quad z_k = x_1 + y_1, & \quad f_k = g_1 h_1; \\ k = 2, & \quad z_k = x_1 + y_2 = x_2 + y_1, & \quad f_k = g_1 h_2 + g_2 h_1; \\ k = m + n - 2, & \quad z_k = x_{m-1} + y_n = x_m + y_{n-1}, & \quad f_k = g_{m-1} h_n + g_m h_{n-1}. \end{aligned}$$

Ein Beispiel ist in Abb. 3.9 (Programm 3.9) gegeben.

```

%-----
%   MATLAB program 3.9: convolution
%   -----
gx = [0 10 20 20 10 0 0 0 0 0 40 0]; x = 1:length(gx);
hy = [ 1 3 1]; y = 1:length(hy); % odd # of channels
y = y - mean(y); hy = hy/sum(hy);
fz = conv(gx,hy); z = (min(x)+min(y)):(max(x)+max(y));
subplot(121)
plot(x,gx,'r*',x,gx,'r-'), hold on
axis([min(z) max(z) 0 max(gx)]), axis('square')
plot(z,fz,'go',z,fz,'g-'), grid
hy = [ 1 3 6 6 3 1]; y = 1:length(hy); % even # of channels
y = y - mean(y); hy = hy/sum(hy);
fz2 = conv(gx,hy); z2 = (min(x)+min(y)):(max(x)+max(y));
plot(z2,fz2,'b+',z2,fz2,'b--'), xlabel('x (z)')
ylabel('g(x) (*), f1(z) (o), f2(z) (+) [o/o]'),
hold off, subplot(122)
plot([0 4],[0 15],'w-'), axis([0 4 -2 15]), axis('square')
for i = 1:length(x), text(1.1,x(i)-0.6, num2str(gx(i))), end
for i = 1:length(z), text(2.1,z(i)-0.6, num2str(fz(i))), end
for i = 1:length(z2), text(3.1,z2(i)-0.6, num2str(fz2(i))), end
xlabel('1: g(x), 2: f1(z), 3: f2(z) [o/o]')
ylabel('x (z)'), grid

%   convolution in C or PASCAL
%   -----
m = length(x); n = length(y); fz3 = zeros(1,m+n-1);
for k = 1:m+n-1
    for i = max(1,k-n+1):min(k,m)
        fz3(k) = fz3(k) + gx(i)*hy(k-i+1);
    end
end
disp(' Z MATLAB C/PASCAL'), disp([z2' fz2' fz3'])
%-----

```

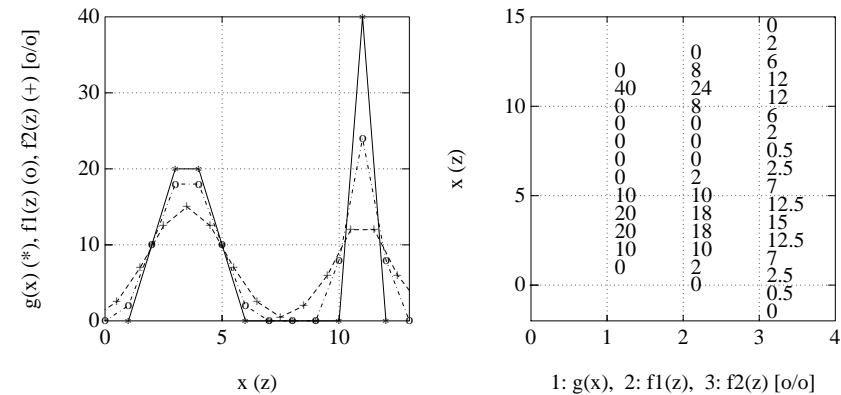


Abbildung 3.9: Faltung einer Verteilung $g(x)$ mit zwei verschiedene Auflösungsfunktionen ($y_1 = (-1, 0, +1)$ mit $h_1(y_1) = (1, 3, 1)/5$ und $y_2 = (-2.5, -1.5, -0.5, +0.5, +1.5, +2.5)$ mit $h_2 = (1, 3, 6, 6, 3, 1)/20$). Die resultierenden Verteilungen sind $f_1(z)$, bzw. $f_2(z)$.

In diesen Beispielen ist der Schwerpunkt der Auflösungsfunktion $h(y)$ gleich Null. Damit wird die ursprüngliche Verteilung $g(x)$ nur verbreitert, aber nicht verschoben. Wenn die x -Werte ganze Zahlen sind (z.B. die Nummerierung der Bins $x_i = i$), dann sind die y -Werte und damit auch die z -Werte ganzzahlig für n ungerade und halbzahlig für n gerade. In MATLAB ist die Faltung $g(x)$ und $h(y)$: $fz = \text{conv}(gx,hy)$. Zum Vergleich ist in Programm 3.9 auch die direkte Programmierung der Gl. 3.31 gegeben, wie man sie sinnemäss in C oder PASCAL schreiben würde.

*Faltung zweier Poisson-Verteilungen

Die Faltung zweier Poisson-Verteilungen mit Mittelwert μ bzw. λ ist eine Poisson-Verteilung mit Mittelwert $\mu + \lambda$. Dies ist aus allgemeinen Überlegungen sofort klar, wie folgendes Beispiel zeigt. Die totale Zählrate einer radioaktiven Probe, die aus zwei α -instabilen Isotopen besteht, zeigt eine Poisson-Verteilung. Auch die Zählraten der beiden Isotopen, die man auf Grund der α -Energien trennen kann, haben je eine Poisson-Verteilung. Ein anderes Beispiel ist die Addition von zwei Bins eines Histogramms. Der mathematische Beweis folgt aus dem allgemeinen Formalismus

(Gl. 3.28: $x_i, y_j = 0 \dots \infty, i, j = 1 \dots \infty$). Mit den Verteilungen (Gl. 2.7)

$$g_i = P(i-1; \mu) = \frac{\mu^{i-1} e^{-\mu}}{(i-1)!} \quad h_j = P(j-1; \lambda) = \frac{\lambda^{j-1} e^{-\lambda}}{(j-1)!} \quad (3.32)$$

und der Binomialentwicklung

$$(\mu + \lambda)^{k-1} = \sum_{l=0}^{k-1} \frac{(k-1)!}{l!(k-1-l)!} \mu^l \lambda^{k-1-l} = (k-1)! \sum_{i=1}^k \frac{\mu^{i-1} \lambda^{k-i}}{(i-1)!(k-i)!}$$

folgt aus Gl. 3.31

$$f_k = \sum_{i=1}^k g_i h_{k+1-i} = \sum_{i=1}^k \frac{\mu^{i-1} \lambda^{k-i} e^{-(\mu+\lambda)}}{(i-1)!(k-i)!} = \frac{(\mu + \lambda)^{k-1} e^{-(\mu+\lambda)}}{(k-1)!} = P(k-1; \mu + \lambda) \quad (3.33)$$

mit $k = 1 \dots \infty$

***Numerische Faltung kontinuierlicher Verteilungen**

Bei der numerischen Faltung von kontinuierlichen Verteilungen werden die Verteilungen an endlich vielen Stützpunkte berechnet. Als Beispiel ist in Abb. 3.10 (Program 3.10) die Faltung einer Exponentialverteilung mit einer Gauss-Verteilung gegeben. Die kontinuierlichen Verteilungen werden in guter Näherung durch diskrete Verteilungen beschrieben, wenn die Binbreite $\Delta \ll \tau$ und $\Delta \ll \sigma$ ist

$$g(x) = \frac{1}{\tau} \exp(-x/\tau) \Rightarrow g_i = \frac{\Delta}{\tau} \exp(-x_i/\tau)$$

$$h(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/2\sigma^2) \Rightarrow h_j = \frac{\Delta}{\sqrt{2\pi}\sigma} \exp(-y_j^2/2\sigma^2)$$

Im Beispiel wurde die Exponentialverteilung statt auf Eins auf die Anzahl der Monte-Carlo-Ereignisse normiert.

```

%-----
%   MATLAB program 3.10: numeric and Monte Carlo convolution
%   -----
bin = 0.1;
Nevents = 800;
sig = 0.7;
x = bin/2:bin:8-bin/2;
y = -2+bin/2:bin:2-bin/2;

binMC = 5*bin;
Norm = Nevents*binMC;
tau = 2.0;
gx = Norm*exp(-x/tau)/tau;
hy = bin*gauss(y,0,sig);
    
```

```

z = min(x)+min(y):bin:max(x)+max(y);
plot(x,gx,'r--')
axis([min(z) max(x) 0 Norm/tau]),
hold on,
plot(z,fz,'g-'),
grid,

fz = conv(gx,hy);
plot(y,0.5*Norm*hy/bin,'b-.')
plot([0 0],[0 Norm/tau],'r--')

%   Monte Carlo convolution
%   -----
xMC = randexp(tau,Nevents,1);
yMC = sig*randn(Nevents,1);
zMC = xMC + yMC;
fz = hist(zMC,z);
plot_data([z' fz'],'k')
xlabel('z'),
ylabel('dN/dz')
%-----
    
```

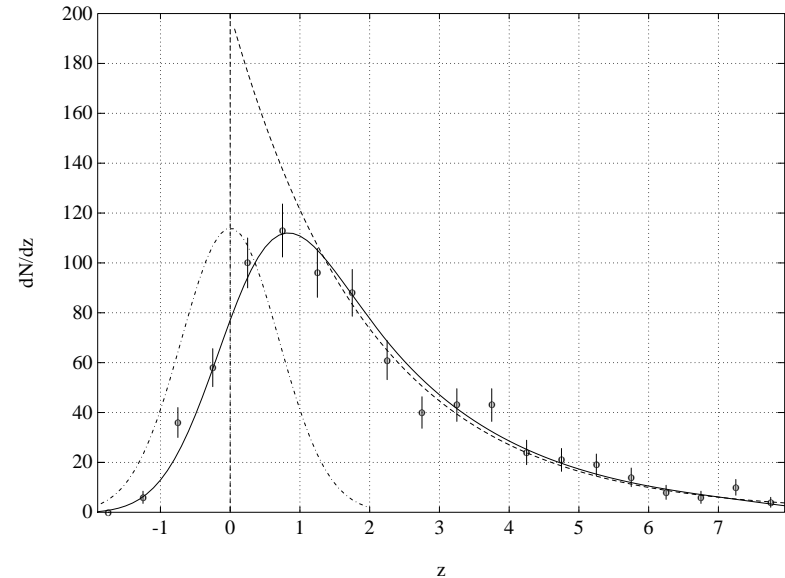


Abbildung 3.10: Numerische Faltung einer Exponentialverteilung (- -) mit einer Gauss-Verteilung (-.). Die resultierende Verteilung (—) ist mit einer Monte-Carlo-Faltung (Histogramm) verglichen.

3.3.2 Monte-Carlo-Faltung; Zentraler Grenzwertsatz

In der Monte-Carlo-Faltung werden die Variablen x_i und y_i paarweise entsprechend ihren Verteilungen $g(x)$ bzw. $h(y)$ gewürfelt und addiert

$$z_i = x_i + y_i \tag{3.34}$$

Durch histogrammieren der z_i erhält man die gefaltete Verteilung $f(z_k)$. Ein Beispiel ist in Abb. 3.10 (Program 3.10) gegeben. Die Monte-Carlo-Faltung (Monte-Carlo-Simulation) wird in der Teilchenphysik vor allem benützt, um der Detektorresponz auf physikalische Ereignisse zu berechnen.

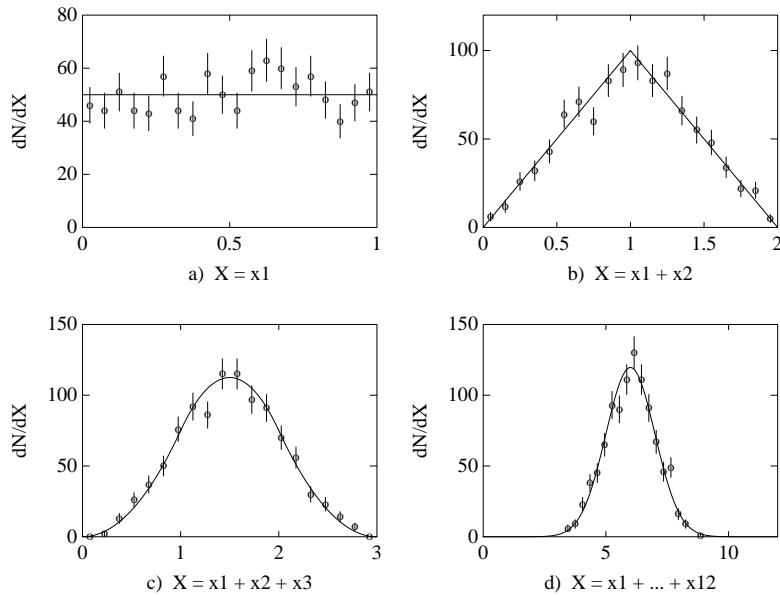


Abbildung 3.11: Monte-Carlo-Faltung von N Gleichverteilungen, jede mit 1000 Ereignisse. (a) $N = 1$, Gleichverteilung; (b) $N = 2$, Dreiecksverteilung; (c) $N = 3$, Verteilung 3.44; (d) $N = 12$, Gauss-Verteilung.

Ein weiteres Beispiel ist die Demonstration des zentralen Grenzwertsatzes. **Der zentrale Grenzwertsatz** (“Central Limit Theorem”):

Ist

$$X = \sum_{i=1}^N x_i \tag{3.35}$$

die Summe von N unabhängigen Variablen x_i , wobei jede aus eine beliebigen Verteilung mit Mittelwert μ_i und Varianz σ_i^2 genommen ist, so ist X für $N \rightarrow \infty$ normal verteilt

$$f(X) \propto \exp \left[-(X - \mu)^2 / 2\sigma^2 \right] \tag{3.36}$$

mit Mittelwert und Varianz

$$\mu = \sum_{i=1}^N \mu_i \quad \sigma^2 = \sum_{i=1}^N \sigma_i^2 \tag{3.37}$$

Das ist der Grund, weshalb die Gauss-Verteilung eine so wichtige Rolle spielt.

Statt eines Beweis (siehe z.B. [Barl 89] oder [Bran 99]) wollen wir als Demonstration die Monte-Carlo-Faltung von Gleichverteilungen betrachten. Jede Variable x_i ist also gleich verteilt ($0 \leq x_i \leq 1$) mit Mittelwert $\mu_i = 1/2$ und Varianz $\sigma_i^2 = 1/12$. In diesem Fall sind die Variablen x_i alle der gleichen Verteilung entnommen. Für $N = 1$ erhalten wir die ursprüngliche Gleichverteilung (Abb. 3.11.a) und für $N = 2$ eine Dreiecksverteilung (Abb. 3.11.b). Für $N = 3$ hat die Verteilung schon eine Ähnlichkeit mit einer Gauss-Verteilung (Abb. 3.11.c) und für $N = 12$ ist sie in guter Näherung eine Gauss-Verteilung mit $\mu = 6$ und $\sigma = 1$ (Abb. 3.11.d).

3.3.3 Kontinuierliche Verteilungen

Die Faltung kontinuierlicher Verteilungen ist identisch mit der Faltung diskreter Verteilungen im Grenzfall $\Delta \rightarrow 0$, $m, n \rightarrow \infty$. Statt Gl.3.28 haben wir jetzt

$$x_{min} \leq x \leq x_{max} \quad \text{und} \quad y_{min} \leq y \leq y_{max} \tag{3.38}$$

Das Intervall der z -Werte ist damit

$$z_{min} \leq z \leq z_{max} \quad \text{mit} \quad z_{min} = x_{min} + y_{min}, \quad z_{max} = x_{max} + y_{max}. \tag{3.39}$$

Die Summe 3.31 geht über in ein Faltungsintegral

$$f(z) = \int_{x_a}^{x_b} g(x) h(z - x) dx = \int_{y_a}^{y_b} g(z - y) h(y) dy \tag{3.40}$$

mit den Integrationsgrenzen

$$\begin{aligned} x_a &= \max(x_{\min}, z - y_{\max}), & x_b &= \min(x_{\max}, z - y_{\min}) \\ y_a &= \max(y_{\min}, z - x_{\max}), & y_b &= \min(y_{\max}, z - x_{\min}) \end{aligned} \quad (3.41)$$

Faltung von Gleichverteilungen

Als erstes Beispiel berechnen wir die Faltung zweier Gleichverteilungen

$$g(x) = 1 \quad \text{für } 0 \leq x \leq 1; \quad h(y) = 1 \quad \text{für } 0 \leq y \leq 1 \quad (3.42)$$

Das Intervall der z -Werte ist also $0 \leq z \leq 2$ und damit $x_a = \max(0, z - 1)$ und $x_b = \min(1, z)$. Die resultierende Verteilung ist (Abb. 3.11b)

$$\begin{aligned} f(z) &= \int_{x_a}^{x_b} dx = z \quad \text{für } 0 \leq z \leq 1 \\ &= 2 - z \quad \text{für } 1 \leq z \leq 2 \end{aligned} \quad (3.43)$$

Diese Verteilung noch einmal mit einer Gleichverteilung $(0,1)$ gefaltet ergibt die Verteilung in Abb. 3.11c

$$\begin{aligned} f(u) &= u^2/2 && \text{für } 0 \leq u \leq 1 \\ &= (-2u^2 + 6u - 3)/2 && \text{für } 1 \leq u \leq 2 \\ &= (u - 3)^2/2 && \text{für } 2 \leq u \leq 3 \end{aligned} \quad (3.44)$$

*Exponentialverteilung und Gauss-Verteilung

Als nächstes Beispiel betrachten wir die Faltung einer Exponentialverteilung mit einer Gauss-Verteilung. Dies kommt in der Praxis oft vor, z.B. bei der Messung der Lebensdauer eines radioaktiven Kerns mit einer endlichen Zeitaufösung der Messapparatur (VP-Versuch Positronium). Die primären Verteilungen sind

$$g(x) = \lambda \exp(-\lambda x) \quad (3.45)$$

und

$$h(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/2\sigma^2) \quad (3.46)$$

mit $0 \leq x \leq \infty$ und $-\infty \leq y \leq \infty$. Das Faltungsintegral ist ($x_a = \max(0, -\infty) = 0$, $x_b = \min(\infty, \infty) = \infty$ oder $y_a = \max(-\infty, -\infty) = -\infty$, $y_b = \min(\infty, z) = z$)

$$\begin{aligned} f(z) &= \frac{\lambda}{\sqrt{2\pi}\sigma} \int_0^\infty \exp[-\lambda x - (z - x)^2/2\sigma^2] dx \\ &= \frac{\lambda}{\sqrt{2\pi}\sigma} \int_{-\infty}^z \exp[-\lambda(z - y) - y^2/2\sigma^2] dy \end{aligned} \quad (3.47)$$

Mit einer Variablentransformation

$$t = \frac{y - \lambda\sigma^2}{\sqrt{2}\sigma} \quad dy = \sqrt{2}\sigma dt$$

erhält man

$$f(z) = \frac{\lambda}{\sqrt{\pi}} \exp[-\lambda(z - \lambda\sigma^2/2)] \int_{t_a}^{t_b} \exp(-t^2) dt$$

mit den Integrationsgrenzen $t_a = -\infty$ ($y = -\infty$) und $t_b = (z - \lambda\sigma^2)/\sqrt{2}\sigma$ ($y = z$). Das Integral kann man durch die Fehlerfunktion ("error function")

$$\operatorname{erf}(u) \equiv \frac{2}{\sqrt{\pi}} \int_0^u \exp(-t^2) dt = \frac{2}{\sqrt{\pi}} \int_{-\infty}^u \exp(-t^2) dt - 1$$

ersetzen

$$f(z) = \frac{\lambda}{2} \exp[-\lambda(z - \lambda\sigma^2/2)] \left[1 + \operatorname{erf}\left(\frac{z - \lambda\sigma^2}{\sqrt{2}\sigma}\right) \right] \quad (3.48)$$

In Abb. 3.12 (Programm 3.11) ist diese analytische Faltung, normiert auf der Zahl der Monte-Carlo-Ereignisse, mit der Monte-Carlo-Faltung verglichen. Beide Verteilungen sind miteinander in Übereinstimmung. Der χ^2 -Wert von 17.0 entspricht bei 18 Freiheitsgraden einer Wahrscheinlichkeit von 52.6 %. Die analytische Faltung ergibt dasselbe Ergebnis wie der numerische Faltung (Abb. 3.10).

```

%-----
%   MATLAB program 3.11: analytic and Monte Carlo convolution
%   -----
load MCdata22,                               logplot_data(MCdata22)
binMC = MCdata22(2,1) - MCdata22(1,1);      Nevents = sum(MCdata22(:,2));
Norm = Nevents*binMC;                       lambda = 1/2.0;
sig = 0.7;                                   bin = 0.1;
z = -2+bin/2:bin:8-bin/2;                   fz = EGconv(z,lambda,sig);
semilogy(z, Norm*fz, 'g-'),                 grid

```

```

xlabel('z'),                ylabel('dN/dz')
x = MCdata22(:,1);        y = MCdata22(:,2);
x = x(y > 0);            y = y(y > 0);
fx = EGconv(x,lambda,sig); w = 1 ./ y;
chisq = sum(w.*(y - Norm*fx).^2), p = Pchisqr(length(x)-1,chisq)

```

```

%-----
function f = EGconv(x,lambda,sig)
f = lambda*exp(-lambda*x + (lambda*sig)^2/2)/2;
f = f.*(1 + erf((x-lambda*sig^2)/(sqrt(2)*sig)));
%-----

```

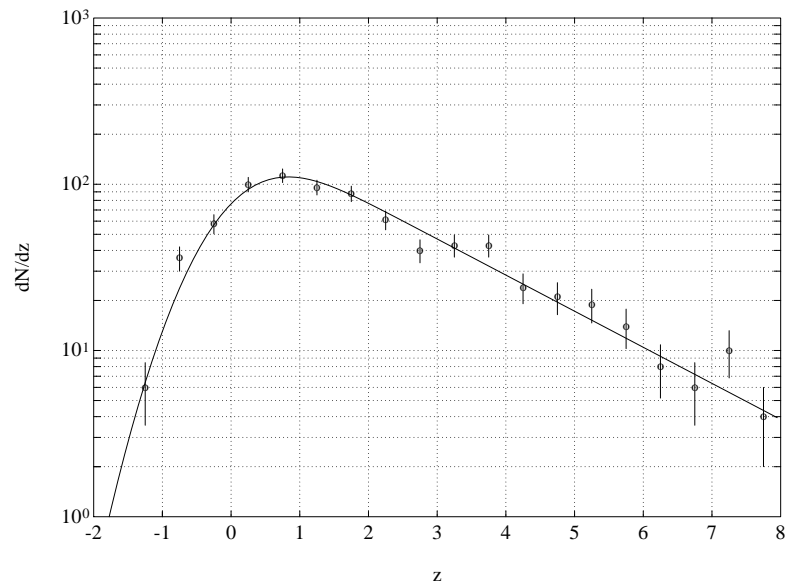


Abbildung 3.12: Analytische Faltung einer Exponentialverteilung mit einer Gauss-Verteilung (—) verglichen mit einer Monte-Carlo-Faltung (Histogramm).

*Faltung zweier Gauss-Verteilungen

Auch das Faltungsintegral für zwei Gauss-Verteilungen

$$g(x) = \frac{1}{\sqrt{2\pi} \sigma_x} \exp[-(x - x_0)^2/2\sigma_x^2], \quad h(y) = \frac{1}{\sqrt{2\pi} \sigma_y} \exp[-(y - y_0)^2/2\sigma_y^2]$$

kann man leicht berechnen. Das Resultat ist eine Gauss-Verteilung mit dem Mittelwert $z_0 = x_0 + y_0$ und der Varianz $\sigma_z^2 = \sigma_x^2 + \sigma_y^2$

$$\begin{aligned}
 f(z) &= \frac{1}{2\pi \sigma_x \sigma_y} \int_{-\infty}^{\infty} \exp[-(x - x_0)^2/2\sigma_x^2 - (z - x - y_0)^2/2\sigma_y^2] dx \\
 &= \frac{1}{\sqrt{2\pi} \sigma_z} \exp[-(z - z_0)^2/2\sigma_z^2]
 \end{aligned}$$

Dies entspricht dem Fehlerfortpflanzungsgesetz bei die Addition zweier Variablen.